IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:		Atty. Docket No.:	003797.00621
Randall E. AULL, et al.			
Serial No.:	10/662,428	Group Art Unit:	2629
Filed:	September 16, 2003	Examiner:	Tammy T. Pham
For:	METHOD FOR PROCESSING DATA QUANTIFYING FORCE APPLIED TO ONE OR MORE KEYS OF A COMPUTER KEYBOARD	Confirmation No.	2075

APPEAL BRIEF UNDER 37 CFR 41.37

U.S. Patent and Trademark Office
Customer Service Window
Mail Stop - Appeal
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

This is an Appeal Brief in support of Appellants' January 4, 2007 Notice of Appeal. Appeal is taken from the final Office Action mailed August 29, 2006. Please charge any necessary fees in connection with this Appeal Brief to our Deposit Account No. 19-0733.

REAL PARTY IN INTEREST

37 C.F.R. § 41.37(c)(1)(i)

The owner of this application, and the real party in interest, is Microsoft Corporation.

RELATED APPEALS AND INTERFERENCES

37 C.F.R. § 41.37(c)(1)(ii)

There are no related appeals or interferences.

STATUS OF CLAIMS
37 C.F.R. § 41.37(c)(1)(iii)

Claims 1-5, 7-27 and 29-44 are pending and rejected. Claims 6 and 28 have been canceled. Appellants hereby appeal the rejection of claims 1-5, 7-27 and 29-44.

STATUS OF AMENDMENTS
37 C.F.R. § 41.37(c)(1)(iv)

No amendments were submitted in response to the August 29, 2006 Final Office Action (hereinafter "Final Office Action"). All previous amendments were entered.

SUMMARY OF CLAIMED SUBJECT MATTER
37 C.F.R. § 41.37(c)(1)(v)

In making reference herein to various embodiments in the specification text and drawings to explain the claimed invention, Appellants do not intend to limit the claims to those embodiments. All references to the specification and drawings are illustrative unless otherwise explicitly stated.

In many prior art keyboards, a given key can only have two states; the key is either pressed (i.e., it is "down") or it is not pressed (i.e., it is "up"). Specification at page 2, at lines 8-10 (para. [05]). Because of this, a single key can only generate two values. Id. at line 10 (para. [05]). A force-sensing key, on the other hand, can generate (or cause the generation of) a range of values corresponding to the amount of force exerted on the key. Id. at line 11-12 (para. [05]). Pressing the key lightly may generate one signal, pressing slightly harder may generate another signal, pressing even harder may generate a third signal, etc. Id. at lines 12-14 (para. [05]). The additional signals can then be given meanings related to a character or function assigned to the key. Id. at lines 14-15 (para. [05]). A keyboard having one or more force-sensing keys presents many data processing challenges, however. For example, an operating system able to process key force data would likely be used in combination with non-force-sensing keyboards. Id. at lines 19-21 (para. [06]). Accordingly, compatibility with both types of keyboards would be highly advantageous. Id. at lines 21-22 (para. [06]). Similarly, not all software applications may be able to use the additional data provided by a force-sensing keyboard, and key force data

should be processed in a manner compatible with applications that do not use key force data. Id. at lines 22-25 (para. [06]).

The invention of independent claim 1 provides a method of processing data received from a force-sensing keyboard having a plurality of keys. The invention of independent claim 23 is directed to a computer-readable medium with instructions for performing steps substantially identical to the steps of claim 1 (the only difference being that certain recitations found in the claim 1 preamble are moved to the body of claim 23). Those steps include receiving keyboard data sets reporting, for keys of the plurality pressed by a keyboard user, key force data and key identification data. Id. at page 6, lines 8-15 (para. [25]) and lines 23-27 (para. [26]); Fig. 1 (ref. character 2); Fig. 2 (ref. characters 43 and 45). The steps also include determining whether key force data in a keyboard data set updates key force data corresponding to a previously-reported key press for a key that remains pressed. Id. at page 3, lines 8-10 (para. [07]). Compare Id. at page 11, lines 1-3 (para. [34])(no force update indicator in message) and Fig. 5 ("keyboard data message") with page 13, line 18 through page 14, line 4 (para. [39])(different action taken based on force update indicator) and Fig. 7 ("keyboard data message"). The steps further include generating first type keyboard data messages containing force updates based on updated key force data, key identifiers for the keys associated with the updated key force data, and force update indicators. Id. at page 3, lines 10-13 (para. [07]); page 13, lines 18-23; Fig. 7 (message in "application 20 message queue"). An additional step comprises generating second type keyboard data messages identifying initially pressed keys and forces applied to the initially pressed keys. Id. at page 3, lines 13-15 (para. 07]); page 11, lines 19-23 (para. [35]); Fig. 5 (messages in "application 20 message queue" and "application 30 message queue"). A further step includes generating, at a repeat rate based on key force data for a key held pressed by a user, a third type keyboard data message indicating the held key has been pressed. Id. at page 12, line 23 through page 13, line 10 (paras. [37] & [38]); page 17, lines 16-19 (para. [47]).

The invention of independent claim 14 also provides a method of processing data received from a keyboard having one or more force-sensing keys. The invention of independent claim 36 is directed to a computer-readable medium with instructions for performing steps substantially identical to the steps of claim 14. Those steps include receiving a keyboard data set

reporting, for multiple keys pressed by a keyboard user, key force data and key identification data. Id. at page 3, lines 18-20 (para. [08]); page 6, lines 18-27 (para. [26]); page 7, lines 14-20 (para. [27]); Fig. 2 (ref characters 41, 43 and 45); Fig. 3 (first, third through fifth and ninth through twelfth slots of ref. character 40). The key identification data is parsed into an ordered list of key identifiers, and the key force data is parsed into an ordered list of key force values. Id. at page 3, lines 20-21 (para. [08]; page 8, lines 1-2 (para. [28]); Fig. 1 (ref. character 10). The steps further include associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the keyboard data set. Id. at page 3, lines 22-23 (para. [08]); page 7, lines 14-19 (para. [27]); page 8, lines 2-4 (para. [28]); Fig. 1 (ref. character 12); Fig. 3 (first and third through fifth slots of ref. char. 40 identify keys; ninth through twelfth slots contain force values (or a null indicator) for identified keys).

The invention of independent claim 17 is also directed to a method of processing data received from a keyboard having one or more force-sensing keys. The invention of independent claim 39 is directed to a computer-readable medium with instructions for performing steps substantially identical to the steps of claim 17. Those steps include receiving a registration from a first application program requesting keyboard input data and key force data. Id. at page 3, lines 24-25 (para. [09]; page 9, lines 25-26 (para. [32]); Fig. 4 (ref. char. 20 and dashed box in ref. char. 16 for application 20). Another registration is received from a second application program requesting keyboard input data but not requesting key force data. Id. at page 3, lines 25-26 (para. [09]; page 10, lines 9-10 (para. [32]); Fig. 4 (ref. char. 30 and dashed box in ref. char. 16 for application 30). The steps further include receiving keyboard data messages identifying keys that have been pressed by a user and containing force values for forces applied to the pressed keys. Id. at page 3, line 27 - page 4, line 2 (para. [09]); page 10, line 17 - page 11, line 1 (para. [34]); page 13, lines 18-19 (para. [39]; Fig. 5 ("keyboard data message"); Fig. 7 ("keyboard data message"). A first keyboard input message is generated, and identifies a first pressed key and contains the force value for the first pressed key. Id. at page 4, lines 2-3 (para. [04]); page 11, lines 15-23 (para. [35]); Fig. 5 (messages in "application 20 message queue" and "application 30 message queue"). A second keyboard input message is also generated, and identifies a second pressed key and contains the force value for the second pressed key and a force update indicator.

Id. at page 4, lines 2-5 (para. [09]); page 13, lines 20-23 (para. [39]); Fig. 7 (message in "application 20 message queue").

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL
37 C.F.R. § 41.37(c)(1)(vi)

Claims 1, 5, 8-12, 14, 23, 27, 28, 30-34 and 36 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,675,329 to Barker et al. (hereinafter "Barker") in view of what appears to be an English translation of the abstract from Japanese Patent Publication No. 05-11914 (hereinafter "Tanaka").

Claims 2-4, 7, 13, 15-21, 24-26, 29, 35 and 37-43 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Barker.

Claims 22 and 44 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Barker in view of U.S. Patent No. 5,576,734 to Danielle et al. (hereinafter "Danielle").

ARGUMENT
37 C.F.R. § 41.37(c)(1)(vii)

Applicants note at the outset that the Examiner rejects independent claims 1, 14, 23 and 36 based on Barker in combination with Tanaka, but then rejects dependent claims 2-4, 7, 13, 15, 16, 24-26, 29, 35, 37 and 38 based solely on Barker. Final Office Action at page 2, lines 13-15 and page 7, lines 16-17. So as to comply with Rule 41.37(c)(1)(vii), Applicants include separate headings for each ground of rejection, with appropriate sub-headings for separately-argued claims, based on the manner in which the Examiner has grouped the rejections.

Applicants also note that any arguments regarding Tanaka made herein rely on the version of Tanaka provided with the Final Office Action and on a machine-generated translation of Tanaka obtained from the Japanese Patent Office web site. A copy of the machine-generated translation is being filed in an IDS submitted with this brief. Applicants have not otherwise procured a translation of Tanaka.

Claims 1, 5, 8-12, 14, 23, 27, 30-34 and 36 Are Patentable Over Barker In View Of Tanaka.

Claims 1, 8-10, 12, 23, 30-32 and 34

Independent claim 1 recites "generating first type keyboard data messages containing force updates based on updated key force data, key identifiers for the keys associated with the updated key force data, and force update indicators." Thus, recited first type keyboard data message includes three separate pieces of data: (1) a force update, (2) a key identifier for a key associated with the updated force data, and (3) a force update indicator. Barker does not describe or suggest such a message.

The Examiner does not explain how Barker purports to teach a keyboard data message including the three required pieces of information. Instead, page 3 of the Final Office Action simply points to reference character 160 of Barker Fig. 2 and to Barker column 4, lines 1-5. Barker Fig. 2 is a flow chart and is reproduced below for convenience.

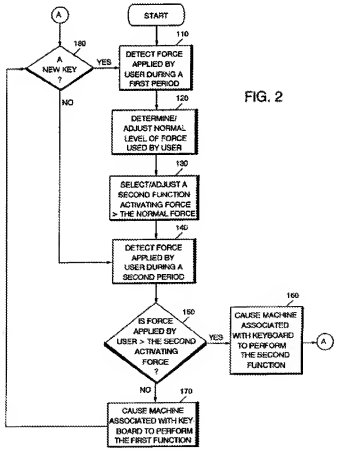

FIG. 2

The flow chart of Barker Fig. 2 generally describes an algorithm, taking place inside of a "microcontroller 18" and other keyboard components, for assigning "key scan data bytes" based on the amount of force applied by a user. In particular, Barker's microcontroller 18 is situated between a "computer keyboard [11] having an X-Y matrix of momentarily depressed key switches having a conventional QWERTY configuration" and a "system unit 24 of a computer." Barker Fig. 1; col. 2, lines 44 through col. 3, line 20. The microcontroller receives a signal representing pressure exerted on a key (or the entire keyboard); based on that signal, the microcontroller determines what "key scan data byte" it should send to the computer. Id. Microcontroller 18 may be housed within keyboard 18. Col. 3, lines 42-44. Barker does not teach microcontroller 18 (or a keyboard in which it is housed) sending force data to a computer. Instead, Barker describes using the force data to determine which key scan data byte will be sent to a computer. The only messages transmitted *from* microcontroller 18 *to* the computer are the key scan data bytes.

Nothing about block 160 in Barker Fig. 2 or the passage at column 4, lines 1-5 even hints at a message having (1) a force update, (2) a key identifier for a key associated with the updated force data, and (3) a force update indicator. Barker makes clear that block 160 ("cause machine associated with keyboard to perform the second function") consists of sending to a computer a key scan data byte that was "retrieve[d]" based on pressure applied to that key. Barker col. 3, lines 6-13.

Similarly, Barker does not teach that anything occurring within microcontroller 18 (i.e., prior to sending a "key scan data byte" to a computer) includes a message having the required three pieces of information. Even if steps (or transitions between steps) in the algorithm of Barker Fig. 2 could be considered to suggest "messages," Barker does not teach or suggest that any single one of those steps or transitions includes a message having all three recited features: a force update based on updated key force data, a key identifier and a force update indicator.

Tanaka similarly fails to teach or suggest a message having a force update based on updated key force data, a key identifier and a force update indicator. Thus, and even if the teachings of Barker could properly be modified based on Tanaka, claim 1 is allowable.

Claim 1 is also allowable based on the recitation of "automatically generating, at a repeat rate based on key force data for a key held pressed by a user, a third type keyboard data message indicating the held key has been pressed." The Examiner concedes that this feature is not taught by Barker, and relies on Tanaka to assert that "[i]t would have been obvious to one with ordinary skill in the art at the time the invention was made to combine the key force data dependent repeat rate as taught by Tanaka with the key force sensitive keyboard of Barker in order to 'improve key operability by controlling the automatic repeat speed of keys on a keyboard for which automatic key repeat control is executed." Final Office Action at page 3 (missing quotation mark in original). Applicants respectfully note that this argument ignores substantial differences between Barker and Tanaka, and thus fails to provide a sufficient motivation for the proposed combination. Specifically, Barker and Tanaka utilize key force data for different purposes. Barker uses that data to choose between two possible key scan data bytes assigned to a particular key (e.g., lower case letter vs. upper case letter). Tanaka uses the force data to decide how frequently a data value assigned to a key is output. Neither Barker nor Tanaka suggests using key force data for both functions. More important, modifying Barker based on Tanaka would change the principle in which the Barker system operates.

Specifically, Barker is directed to a system that uses a key force to determine which of multiple key scan data bytes is to be output for a particular key. Indeed, the Barker invention "relates to methods of obtaining a second function from keys on a keyboard that have only a first function when struck individually." Barker col. 1, lines 8-10. Tanaka is directed to controlling the rate at which a key output is generated. Combining the teaching of these two references would have required substantial modification to the Barker system, assuming *arguendo* that such a modification would even have been possible without destroying the main purpose of the Barker system. Assume, for example, that the Barker system outputs a lower case letter for a key force less than the "second actuating force" threshold and an upper case letter for a key force above that threshold. If the key force is also used to control repeat rate, does that mean that upper case letters would be repeated and lower case letters would not be repeated (or that upper case letters would be repeated quickly and lower case letters repeated slowly)? Because "the 'suggested combination of references would require a substantial reconstruction and redesign of the

elements shown in [the primary reference] as well as a change in the basic principle under which the [primary reference] construction was designed to operate,' " the teachings of Barker and Tanaka do not support a *prima facie* case of obviousness. See MPEP § 2143.01 VI. (citing In re Ratti, 270 F.2d 810, 813 (CCPA 1959))(bracketed portions in original).

In the December 4, 2006 Advisory Action (hereinafter "Advisory Action"), the Examiner asserted that " '[t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference.... Rather, the test is what the combined teachings of those references would have suggested to those of ordinary skill in the art.' In re Keller, 642 F.2d 413, 425 ... (CCPA 1981)." The Examiner appears to have been quoting MPEP § 2145 III. Notably, the last sentence of § 2145 III also states: "However, the claimed combination cannot change the principle of operation of the primary reference or render the reference inoperable for its intended purpose. See MPEP § 2143.01." As discussed above, any attempt to make the proposed combination would have involved a significant change in the principle of operation of Barker.

For at least the above reasons, claim 1 is allowable. Claim 23 is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 1, and is thus also allowable. Each of claims 8-10, 12, 30-32 and 34 depends from one of claims 1 or 23.

Claims 5 and 27

Claim 5 depends from claim 1, and recites that the step of receiving keyboard data sets comprises receiving a data set having key identification data and key force data for multiple keys. Claim 5 also recites the additional steps of parsing the key identification data in the keyboard data set into an ordered list of key identifiers and parsing the key force data in the keyboard data set into an ordered list of key force values. In rejecting claims 5 and 27, the Examiner relied on Barker Fig. 2 and "column 3, lines 50-20," together with Barker column 4, line 22. Final Office Action at pages 4, 14. However, neither these nor any other portion of Barker teaches or suggests a requirement of claim 5 -- a single data set with key identification and key force data for *multiple* keys, which data set (for multiple keys) is then parsed into an

ordered list of key identifiers and into an ordered list of key force values. Barker instead teaches that key data is received, for one key at a time, in successive loops through the algorithm of Barker Fig. 2.

The Examiner appears to read claim 5 such that there is no requirement for a data set that contains force and identification data for one key while simultaneously containing force and identification data for another key. See Advisory Action at page 2. However, this is at odds with the claim language that requires the data set to be parsed into ordered lists of key identifiers and key force values. Specifically, the antecedent basis for "the key identification data" in the first parsing step is the received key identification data for multiple keys. Similarly, the antecedent basis for "the key force data" in the second parsing step is the received key force data for multiple keys. The parsing steps thus require that there be some separation of the two types of data (key identifiers and force values) from a *received* data set.

For at least these additional reasons, claim 5 is allowable. Claim 27 (which depends from independent claim 23) is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 5, and is thus also allowable. Tanaka also fails to teach the additional features of claims 5 and 27 not taught by Barker, and the Examiner has identified no portion of Tanaka purporting to teach such features.

Claims 11 and 33

Claim 11 depends from claims 1, 8, 9 and 10. As such, claim 11 recites:

> ...wherein said automatically generating a third type keyboard data message comprises mapping a repeat rate to the key force data for the held key[, and] comprising:
>
> storing cumulative key force data; and
>
> based on the stored cumulative key force data, mapping a repeat rate to the force data for the held key[, and]
>
> wherein said mapping is based on a transfer function in which a range of force data values is subdivided into multiple sub-ranges, and wherein each of the sub-ranges is assigned a repeat rate[, and]
>
> wherein the transfer function comprises an *initial group of sub-ranges mapped to gradually increasing repeat rate values* followed by a *group of sub-ranges mapped to sharply increasing repeat rate values* [emphasis added].

A transfer function, as set forth in claim 11 and as described in the specification at page 18, lines 10-23 and in Fig. 11, maps a sub-range of key force values to a repeat rate. At least two of those sub-ranges (the initial group) are mapped to gradually increasing repeat rate values, and at least two sub-ranges (the group that follows the initial group) are mapped to sharply increasing key force values. Although "gradually" and "sharply" are not quantified, it is clear that "sharply" refers to a rate of increase that is higher relative to a rate of increase that is "gradually" increasing.

Neither Barker nor Tanaka teaches or suggests a transfer function as is recited in claim 11. The Examiner relies on Barker column 3, lines 1-20 and on "the constitution of Tanaka" to argue that "[t]he force that the user exerts in pressing the buttons gradually increases in order to reach the second function of that particular button." Putting aside that neither Barker nor Tanaka describes "gradually" increasing button force, the Examiner's argument is off the mark. It is not the gradation of user-applied force that is relevant to claim 11. Instead, claim 11 relates to how key force is converted into a repeat rate. Although Tanaka may indeed teach that "the key repeat speed depends on the pressure sensor of the key" (Final Office Action at page 14) and this may suggest that "there are various rates correlating to various pressures of the user to the keys" (Id.), Tanaka gives no hint at how sharply or gradually any of those "various rates" increases for any particular range of key forces.

Accordingly, and for at least this additional reason, claim 11 is allowable. Claim 33 is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 11, and is thus also allowable for this additional reason.

Claims 14 and 36

Independent claim 14 is directed to a method of processing data received from a keyboard having one or more force-sensing keys. The method includes receiving a keyboard data set reporting, for multiple keys of the plurality pressed by a keyboard user, key force data and key identification data; parsing the key identification data into an ordered list of key identifiers; and parsing the key force data into an ordered list of key force values. As explained above for claims 5 and 27, these features are not taught or suggested by Barker or Tanaka.

Accordingly, claim 14 is allowable. Independent claim 36 is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 14, and is thus also allowable.

Claims 2-4, 7, 13, 15-21, 24-26, 29, 35 and 37-43 Are Patentable Over Barker.

Claims 2 and 24

Claims 2 and 24 depend from claims 1 and 23, respectively, and are thus allowable for the same reasons set forth above in connection with claims 1 and 23. Applicants separately list claims 2 and 24 herein so as to comply with Rule 41.37(c)(1)(vii).

Claims 3, 4, 13, 25, 26 and 35

Claim 3 depends from claim 1 and is allowable for the same reasons set forth above in connection with claim 1. Claim 3 is additionally allowable based on the additional steps recited in claim 3: determining if reported key force data contains a null indicator, and associating a null indicator with a non-force-sensing key.

As indicated in claim 3, a null indicator is used to indicate that a keyboard data set corresponds to a key that does not sense force. An illustrative embodiment consistent with the claim 3 invention is described in the specification at page 7, lines 6-10. The Examiner concedes that Barker fails to teach the features of claim 3 (or 4), but then make the following argument:

> It would have been obvious to have a way to determine if no key was pressed, or more specifically, if in that determining if reported key force data contains a null indicator; and associating a null indicator with a non-force-sensing key {claim 3} or that wherein a null indicator is a zero value for key force data {claim 4} so that the apparatus can detect whether or not a key is in use.
>
> It would have been obvious to one with ordinary skill in the art at the time the invention was made to include that in that determining if reported key force data contains a null indicator; and associating a null indicator with a non-force-sensing key {claim 3} or that wherein a null indicator is a zero value for key force data {claim 4} so that the apparatus can detect whether or not a key is in use to better monitor the keyboard pressure/force (see Barker, column 2, lines 35-36).

Final Office Action at pages 8-9.

Nothing in Barker suggests that non-force-sensing keys would have any kind of force data. Applicants also submit that the above argument for a motivation to have modified Barker is not logical. The purported motivation to have modified Barker is that a null indicator could have been used to detect "whether or not a key is in use." Implicit in this argument is that Barker should have been modified so as to use null force indicators for unpressed keys. However, such a modification would not teach all features of claim 3, which requires that the reported key force data in claim 3 be part of a data set that reports key force data for *pressed* keys. In particular, "reported key force data" in claim 3 refers to the key force data introduced in the first step of claim 1: "receiving keyboard data sets *reporting*, for keys of the plurality *pressed* by a keyboard user, *key force data* and key identification data" (emphasis added).

Accordingly, and for this additional reason, claim 3 is allowable. Claim 4 depends from claim 3 and is also allowable for this additional reason.

Claim 13 recites determining if the key force data for another *held* key contains a null indicator. Thus, for the same reasons set forth above in connection with claim 3, claim 13 is also allowable based on this additional recited feature.

Claims 25, 26 and 35 are directed to computer-readable media having instructions for performing steps substantially identical to those of claims 3, 4 and 13, respectively, and are thus also allowable for the same additional reasons applicable to claims 3, 4 and 13.

Claims 7 and 29

Claims 7 and 29 depend from claims 1 and 23, respectively, and are thus allowable for the same reasons set forth above in connection with claims 1 and 23. Applicants separately list claims 7 and 29 herein so as to comply with Rule 41.37(c)(1)(vii).

Claims 15, 16, 37 and 38

Claims 15 and 16 depend from independent claim 14, and are allowable for the same reasons discussed above in connection with claim 14. Moreover, claim 15 recites the additional step of determining if reported key force data contains a null indicator. As in claim 3 (discussed above), the "reported key force data" in claim 15 must be part of a data set that reports key force

data for *pressed* keys. Specifically, the reported key force data of claim 15 refers to the key force data introduced in the "receiving" step of claim 14. Claim 15 is thus additionally allowable for the same reasons explained above in connection with claim 3. Claim 16 depends from claim 15 and is also allowable for this additional reason.

Claims 37 and 38 are directed to computer-readable media having instructions for performing steps substantially identical to those of claims 15 and 16, respectively, and are thus also allowable for the same additional reasons applicable to claims 15 and 16.

Claims 17-19 and 39-41

The invention of independent claim 17 is also directed to a method of processing data received from a keyboard having one or more force-sensing keys. The method includes the steps of receiving a registration from a first application program requesting keyboard input data and key force data and receiving a registration from a second application program requesting keyboard input data but not requesting key force data.

Barker does not teach either of the "receiving a registration..." steps of claim 17. The Examiner concedes at page 10 of the Final Office Action that Barker fails to teach these steps. The Examiner then asserts that Barker's teachings should have been modified so as to include all features of claim 17. In particular, the Examiner makes the following argument(s):

> [I] However, in Fig. 2 and in column 3, lines 50-20; Barker teaches of the process that has various prompts to take in keyboard data because this has a similar function to the applications as in the claim limitations, the prompts can be seen and viewed as separate applications.
>
> [II] It would have been obvious to one with ordinary skill in the art at the time the invention was made to include application software with the apparatus in order to associate a different task or data with a different application.
>
> [III] Since the apparatus of Barker is able to detect and communicate data information in regards to both (1) which key is pressed (see column 3, line 55) and (2) the key force data (see column 4, line 5); then it would be known in the art to provide an apparatus that is able to only send only one set of data (either the which [sic] key is pressed or the key force data).
>
> [IV] It would have been obvious to one with ordinary skill in the art at the time the invention was made to combine only send [sic] one set of data, such as the keyboard input instead of both the keyboard input and the key force data in order

> to only provide needed data needed by various application to avoid unnecessary steps.

Final Office Action at page 10 (bracketed Roman numerals added for purpose of subsequent reference).

With regard to argument I, the Examiner equates the supposed "prompts" in Barker Fig. 2 and "column 3, lines 50-20" with the requests by the first and second applications of claim 17 for certain types of data. Barker, however, says nothing about "prompts" in the flowchart of Fig. 2 or elsewhere. Apparently recognizing this flaw in argument I, the Examiner stated the following in the Advisory Action: "Fig. 2 of BARKER may not teach explicitly of 'prompts' but the very first step in the flow chart is 'start,' where start can be interpretated [sic] as prompt." The Examiner's subsequent statement is incorrect. The "START" block in Barker Fig. 1 is not a prompt, and Barker does not give any suggestion that it represents a request from an application program for some type of data. The "START" block instead is simply a standard flow chart symbol for a beginning point for the Barker Fig. 2 algorithm. Indeed, all of the blocks in that flowchart simply reflect steps of a single algorithm. Steps of that algorithm are not "applications," much less two separate applications. Even if the individual steps of the algorithm in Barker Fig. 2 could be considered applications, however, Barker in no way suggests that those steps register with other parts of the algorithm and request certain types of data, and even less so in the particular manner claimed.

With regard to argument II, it is not clear whether the Examiner asserts it would have been obvious to include multiple applications in the Barker keyboard, or whether the Examiner asserts it would have been obvious to include multiple applications in a computer to which the Barker keyboard is connected. As to the latter, Applicants agree that the presence of multiple application programs in a computer was known. However, it is clear that Barker does not describe (or even remotely contemplate) the presence of applications that process key force data. Any key force data in Barker remains within the confines of microcontroller 18, and thus is not taught or suggested as being accessible to an application. Instead, Barker appears to assume that the operations carried out in connection with Barker Fig. 2 convert key force data into other (non-force) information understood by all applications. Barker col. 3, lines 5-28 and 42-44

indicate that the algorithm of Barker figure 2 is performed by a microcontroller 18 within a keyboard, with that microcontroller deciding which key scan code is to be sent (by that same microcontroller) to a "system unit 24 of a computer such that the computer is instructed to perform" a corresponding function. Examples of the types of key scan data bytes that the Barker keyboard would select based on key force are found in Barker column 4, lines 28-44 (a lower case letter, TAB function, F1-F12 function, control function or alt function for keys struck with a lesser force, and a BACK-TAB function, shifted F1-F12 function, shifted control function or shifted alt function for keys struck with a greater force). Conspicuously absent is any suggestion that a keyboard would send (or have any need to send) force data to the computer.

Because Barker contemplates determining how to interpret key force data before that data even reaches a computer, there would have been no need or desire for an application in the Barker computer to request key force data (and thus no motivation to modify Barker's teachings). In fact, such a request could not have been fulfilled since the key force data is not taught or suggested as being accessible outside of microcontroller 18. Stated differently, Barker is directed to hardware that determines key force and then selects a key code based on that force; the hardware then sends that key code to the computer. There would be no reason for an application in the computer to request force data itself, as the hardware has already determined what the key force data means.

If the Examiner's argument II is instead meant to assert that it would have been obvious to include multiple applications in the Barker keyboard, then the Examiner has failed to explain why a person of ordinary skill in the art would have been motivated to make such major modifications to the Barker keyboard (or to even explain what those modifications would include).

The Examiner's arguments III and IV are built on a false premise. Barker simply does *not* teach that the described apparatus is able to "detect and communicate data information in regards to both (1) which key is pressed ... and (2) the key force data." Even if it would have been "known in the art to provide an apparatus that is able to only send only one set of data (either the which [sic] key is pressed or the key force data)," there would have been no reason to do so from a device that determines how key force data should be interpreted before

communicating with the computer as a result of the that key force. There would thus have been no motivation to modify the Barker keyboard to transmit force data to a computer.

In the Advisory Action, the Examiner retreated from argument III, stating that "BARKER is not relied upon as to being able to communicate both key identification and key force data" and that "the claim language [of claim 17] never states of the apparatus being able to 'communicate information.' " Applicants agree that claim 17 does not recite a device "able to detect and communicate data information in regards to both (1) which key is pressed ... and (2) the key force data." However, the purported teaching of such a device by Barker was the basis for the Examiner's motivation to modify Barker. With the Examiner's withdrawal of this basis, argument III is now just a bare allegation.

Finally, Applicants feel obliged to comment upon the following statement from page 2 the Advisory Action:

> CLAIM 17 (Page 18 of arguments) recites "receiving a registration from a first application program requesting keyboard input data and key force data; receiving a registration from a second application program requesting keyboard input data but not requesting key force data." In the previous Office Action, BARKER is shown to teach all of the claim limitation and Examiner took Office Notice that it was known in the art to include a first and second application in order to properly carry out the functionality of the apparatus. [underlining added]

The single-underlined portion is contrary to the following statement from page 10 (lines 4-6) of the Final Office Action: "Barker fails to teach of receiving a registration from a first application program requesting keyboard input data and key force data; receiving a registration from a second application program requesting keyboard input data but not requesting key force data." The double-underlined statement is also incorrect. No official notice was taken in either the Final Office Action or in the preceding Office Action mailed March 21, 2006. To the extent the Examiner contends that the statement in the Advisory Action constitutes official notice of some fact, Applicants traverse on the ground that the statement is not sufficiently specific as to what is being noticed.

For at least the reasons set forth above, claim 17 is allowable. Claims 18, 19 and 21 depend from claim 17 and are allowable based on this dependency.

Independent claim 39 and dependent claims 40, 41 and 43 are directed to computer-readable media having instructions for performing steps substantially identical to those of claims 17-19 and 21, respectively, and are thus also allowable.

Claims 20 and 42

Claim 20 depends from claim 17, and recites the step of generating a third keyboard input message identifying a third pressed key and containing the force value for the third pressed key and a force update indicator. Similar to claim 1, claim 20 recites a message that includes three separate pieces of data: (1) an identification of the pressed third key, (2) the force values for that key, and (3) a force update indicator. As explained above in connection with claim 1, Barker does not teach or suggest such a message. Accordingly, claim 20 is also allowable for this additional reason. Claim 42 is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 20, and is thus also allowable for this additional reason.

Claims 22 and 44 Are Patentable Over Barker In View Of Danielle.

Claim 22 depends from claim 17. Danielle fails to teach the above-discussed features of claim 17 not found in Barker, and claim 22 is thus allowable on this basis. Claim 44 is directed to a computer-readable medium having instructions for performing steps substantially identical to those of claim 22, and is thus also allowable.

CONCLUSION

For all of the foregoing reasons, Appellants respectfully submit that the final rejection of claims 1-5, 7-27 and 29-44 is improper and should be reversed.

Respectfully submitted,
BANNER & WITCOFF, LTD.

Dated: March 5, 2007

By: **/H. Wayne Porter/**
H. Wayne Porter
Registration No. 42,084

1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel: (202) 824-3000
Fax: (202) 824-3001

CLAIMS APPENDIX
37 C.F.R. § 41.37(c)(1)(viii)

1. A method of processing data received from a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, the method comprising:

receiving keyboard data sets reporting, for keys of the plurality pressed by a keyboard user, key force data and key identification data;

determining whether key force data in a keyboard data set updates key force data corresponding to a previously-reported key press for a key continuing to be pressed;

generating first type keyboard data messages containing force updates based on updated key force data, key identifiers for the keys associated with the updated key force data, and force update indicators;

generating second type keyboard data messages identifying initially pressed keys and forces applied to the initially pressed keys; and

automatically generating, at a repeat rate based on key force data for a key held pressed by a user, a third type keyboard data message indicating the held key has been pressed.

2. The method of claim 1, wherein the first and second type keyboard data messages have a common data structure.

3. The method of claim 1, further comprising:

determining if reported key force data contains a null indicator; and

associating a null indicator with a non-force-sensing key.

4. The method of claim 3, wherein a null indicator is a zero value for key force data.

5. The method of claim 1, wherein said receiving keyboard data sets comprises receiving a data set having key identification data and key force data for multiple keys, and further comprising:

parsing the key identification data in the keyboard data set into an ordered list of key identifiers;

parsing the key force data in the keyboard data set into an ordered list of key force values; and

associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the data set.

7. The method of claim 1, wherein the first, second and third type keyboard data messages have a common data structure.

8. The method of claim 1, wherein said automatically generating a third type keyboard data message comprises mapping a repeat rate to the key force data for the held key.

9. The method of claim 8, comprising:

storing cumulative key force data; and

based on the stored cumulative key force data, mapping a repeat rate to the force data for the held key.

10. The method of claim 8, wherein said mapping is based on a transfer function in which a range of force data values is subdivided into multiple sub-ranges, and wherein each of the sub-ranges is assigned a repeat rate.

11. The method of claim 10, wherein the transfer function comprises an initial group of sub-ranges mapped to gradually increasing repeat rate values followed by a group of sub-ranges mapped to sharply increasing repeat rate values.

12. The method of claim 1, wherein said automatically generating a third type keyboard message comprises:

determining if a repeat invoke delay has elapsed since the user initially pressed the held key; and

commencing said automatic generation after the repeat invoke delay has elapsed.

13. The method of claim 1, further comprising:

determining if the key force data for another held key contains a null indicator; and

upon determining that the key force data for the other held key contains a null indicator, automatically generating, at a preset rate and after a preset delay, repeating keyboard data messages indicating the other held key has been pressed.

14. A method of processing data received from a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, the method comprising:

receiving a keyboard data set reporting, for multiple keys of the plurality pressed by a keyboard user, key force data and key identification data;

parsing the key identification data into an ordered list of key identifiers;

parsing the key force data into an ordered list of key force values; and

associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the keyboard data set.

15. The method of claim 14, further comprising:

determining if reported key force data contains a null indicator; and

associating a null indicator with a non-force-sensing key.

16. The method of claim 15, wherein a null indicator is a zero value for key force data.

17. A method for processing data received from a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, the method comprising:

receiving a registration from a first application program requesting keyboard input data and key force data;

receiving a registration from a second application program requesting keyboard input data but not requesting key force data;

receiving keyboard data messages identifying keys that have been pressed by a user and containing force values for forces applied to the pressed keys;

generating a first keyboard input message identifying a first pressed key and containing the force value for the first pressed key; and

generating a second keyboard input message identifying a second pressed key and containing the force value for the second pressed key and a force update indicator.

18. The method of claim 17, further comprising:

providing the first keyboard input message to the first and second applications.

19. The method of claim 18, further comprising:

only providing the second keyboard input message to applications requesting key force data.

20. The method of claim 17, wherein the second keyboard input message is provided to the first application, and further comprising:

generating a third keyboard input message identifying a third pressed key and containing the force value for the third pressed key and a force update indicator; and

providing the third keyboard input message to the first application prior to providing a message indicating that the second pressed key is no longer being pressed.

21. The method of claim 17, comprising:

storing the identifier for the last key identified as pressed;

storing the most recently received force value for the last key identified as pressed;

receiving a keyboard data message lacking a force value and indicating that the last key identified as pressed remains pressed; and

generating a keyboard input message identifying the last key identified as pressed and containing the stored force value.

22. The method of claim 17, comprising:

receiving a simulated keyboard data message containing simulated key press data, the simulated key press data identifying an unpressed key and containing simulated key force data for the unpressed key; and

generating a third keyboard input message identifying the unpressed key, indicating a simulated key press, and containing the simulated key force value.

23. A computer-readable medium having stored thereon data representing sequences of instructions which, when executed by a processor, cause the processor to perform steps comprising:

receiving keyboard data sets from a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, wherein the keyboard data sets report, for keys of the plurality pressed by a keyboard user, key force data and key identification data;

determining whether key force data in a keyboard data set updates key force data corresponding to a previously-reported key press for a key continuing to be pressed;

generating first type keyboard data messages containing force updates based on updated key force data, key identifiers for the keys associated with the updated key force data, and force update indicators;

generating second type keyboard data messages identifying initially pressed keys and forces applied to the initially pressed keys; and

automatically generating, at a repeat rate based on key force data for a key held pressed by a user, a third type keyboard data message indicating the held key has been pressed.

24. The computer-readable medium of claim 23, wherein the first and second type keyboard data messages have a common data structure.

25. The computer-readable medium of claim 23, comprising further instructions for performing steps comprising:

determining if reported key force data contains a null indicator; and

associating a null indicator with a non-force-sensing key.

26. The computer-readable medium of claim 25, wherein a null indicator is a zero value for key force data.

27. The computer-readable medium of claim 23, wherein said receiving keyboard data sets comprises receiving a data set having key identification data and key force data for multiple keys, and comprising further instructions for performing steps comprising:

parsing the key identification data in the keyboard data set into an ordered list of key identifiers;

parsing the key force data in the keyboard data set into an ordered list of key force values; and

associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the data set.

29. The computer-readable medium of claim 23, wherein the first, second and third type keyboard data messages have a common data structure.

30. The computer-readable medium of claim 23, wherein said automatically generating a third type keyboard data message comprises mapping a repeat rate to the key force data for the held key.

31. The computer-readable medium of claim 30, comprising further instructions for performing steps comprising:

storing cumulative key force data; and

based on the stored cumulative key force data, mapping a repeat rate to the force data for the held key.

32. The computer-readable medium of claim 30, wherein said mapping is based on a transfer function in which a range of force data values is subdivided into multiple sub-ranges, and wherein each of the sub-ranges is assigned a repeat rate.

33. The computer-readable medium of claim 32, wherein the transfer function comprises an initial group of sub-ranges mapped to gradually increasing repeat rate values followed by a group of sub-ranges mapped to sharply increasing repeat rate values.

34. The computer-readable medium of claim 23, wherein said automatically generating a third type keyboard message comprises:

determining if a repeat invoke delay has elapsed since the user initially pressed the held key; and

commencing said automatic generation after the repeat invoke delay has elapsed.

35. The computer-readable medium of claim 23, comprising further instructions for performing steps comprising:

determining if the key force data for another held key contains a null indicator; and

upon determining that the key force data for the other held key contains a null indicator, automatically generating, at a preset rate and after a preset delay, repeating keyboard data messages indicating the other held key has been pressed.

36. A computer-readable medium having stored thereon data representing sequences of instructions which, when executed by a processor, cause the processor to perform steps comprising:

receiving a keyboard data set from a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, wherein the keyboard data sets report, for multiple keys of the plurality pressed by a keyboard user, key force data and key identification data;

parsing the key identification data into an ordered list of key identifiers;

parsing the key force data into an ordered list of key force values; and

associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the keyboard data set.

37. The computer-readable medium of claim 36, comprising further instructions for performing steps comprising:

determining if reported key force data contains a null indicator; and

associating a null indicator with a non-force-sensing key.

38. The computer-readable medium of claim 38, wherein a null indicator is a zero value for key force data.

39. A computer-readable medium having stored thereon data representing sequences of instructions which, when executed by a processor, cause the processor to perform steps comprising:

receiving a registration from a first application program requesting keyboard input data and key force data;

receiving a registration from a second application program requesting keyboard input data but not requesting key force data;

receiving keyboard data messages identifying keys that have been pressed by a user on a keyboard having a plurality of keys, the plurality of keys including multiple keys having respective characters assigned thereto, the plurality of keys further including one or more force-sensing keys, wherein the keyboard data messages contain force values for forces applied to the pressed keys;

generating a first keyboard input message identifying a first pressed key and containing the force value for the first pressed key; and

generating a second keyboard input message identifying a second pressed key and containing the force value for the second pressed key and a force update indicator.

40. The computer-readable medium of claim 39, comprising further instructions for performing steps comprising:

providing the first keyboard input message to the first and second applications.

41. The computer-readable medium of claim 40, comprising further instructions for performing steps comprising:

only providing the second keyboard input message to applications requesting key force data.

42. The computer-readable medium of claim 39, wherein the second keyboard input message is provided to the first application, and comprising further instructions for performing steps comprising

generating a third keyboard input message identifying a third pressed key and containing the force value for the third pressed key and a force update indicator; and

providing the third keyboard input message to the first application prior to providing a message indicating that the second pressed key is no longer being pressed.

43. The computer-readable medium of claim 39, comprising further instructions for performing steps comprising:

storing the identifier for the last key identified as pressed;

storing the most recently received force value for the last key identified as pressed;

receiving a keyboard data message lacking a force value and indicating that the last key identified as pressed remains pressed; and

generating a keyboard input message identifying the last key identified as pressed and containing the stored force value.

44. The computer-readable medium of claim 39, comprising further instructions for performing steps comprising:

receiving a simulated keyboard data message containing simulated key press data, the simulated key press data identifying an unpressed key and containing simulated key force data for the unpressed key; and

generating a third keyboard input message identifying the unpressed key, indicating a simulated key press, and containing the simulated key force value.

EVIDENCE APPENDIX
37 C.F.R. § 41.37(c)(1)(ix)

NONE

RELATED PROCEEDINGS APPENDIX
37 C.F.R. § 41.37(c)(1)(x)

NONE